

Foldover Effects on Viterbi Decoding

J. K. Omura¹ and S. A. Butman
Communications Systems Research Section

Viterbi decoding of X-band Mariner Venus-Mercury 1973 (MVM'73) spacecraft data using both a hardware Viterbi decoder and a software Viterbi decoder resulted in a significant and previously unexplained difference in decoded bit error rates. This difference is explained by foldover effects which arose when the 6-bit recorded data were reduced to the required 3-bit decoder input data in the hardware Viterbi decoder.

I. The Problem

On January 15, 1974, X-band convolutionally coded telemetry data were transmitted from the MVM'73 spacecraft and recorded at the Goldstone 64-m station. The purpose of the experiment was to use an in-flight spacecraft and an operational Deep Space Station (DSS) to demonstrate that X-band high-data-rate convolutionally coded telemetry data can be reliably transmitted according to the predicted theoretical performance.

The signal transmitted by the MVM'73 spacecraft was a periodic sequence consisting of repetition of bits 1 1 1 0 1 0, which corresponds to a periodic sequence of data bits 1 0 0 convolutionally encoded by a constraint length $\nu = 7$, rate $r = 1/2$ encoder. A hardware Viterbi decoder was employed on the recorded data and the decoded bit error probabilities were computed over a range of decoded bit signal-to-noise ratio E_b/N_0 of approximately 3 dB to 5 dB. Later the same recorded data were decoded using a software decoder that performed

the same Viterbi decoder operations. The hardware decoder experimental results and the software decoder experimental results based on the same X-band convolutionally coded telemetry data are shown in Fig. 1.

At $E_b/N_0 = 4$ dB, which is about the middle value of the experimental signal-to-noise ratio range, we see that the Linkabit decoder experiment had 42 times more bit errors than the predicted theoretical performance while the software decoder bit error rates agreed closely with the predicted theoretical performance. The problem is then to explain the large difference between two decoder experiments where both are apparently performing the same operations on the same X-band data.

II. Coded Data Reduction

The coded bits transmitted by the MVM'73 spacecraft appeared as telemetry sidebands of the X-band carrier. Reception of the signals was accomplished using the Block IV receiver system, and the recorded data consisted of the SSA matched filter output quantized with a sign bit plus 5 magnitude bits.

¹Consultant, University of California at Los Angeles.

The hardware Viterbi decoder accepts only 3-bit quantized (8 levels) data as inputs. With proper quantization spacing, performance with this 3-bit quantization has been shown to be within 0.25 dB in signal-to-noise ratio of the performance achievable with infinite quantization (Ref. 1). This is often referred to as “soft decision” decoding and the coding channel is modeled as a discrete memoryless channel with two input symbols and 8 output symbols. With about 2 dB degradation Viterbi decoding can be done using only the sign bit or one-bit quantized data (Ref. 2). The resulting “hard decision” coding channel is called a binary symmetric channel.

In the experimental results of Fig. 1, both decoders used 3-bit quantized data as inputs. It was discovered that there was an apparently small difference in how the 6-bit recorded data were converted to 3-bit decoder input data for the two experiments. Let $x_1, x_2, x_3, x_4, x_5, x_6$ be the 6 bits of any recorded data sample, where x_1 is the sign bit and the remaining 5 bits are magnitude bits with x_2 being the most significant and x_6 being the least significant. To achieve a good 3-bit quantization spacing, the least significant bit required was x_4 . Hence x_5 and x_6 were not used. There remained the problem of reducing 4-bit data x_1, x_2, x_3, x_4 to 3 bits.

In the 4-bit data the bits x_1, x_2, x_3, x_4 correspond to one of 16 uniformly spaced amplitudes as follows:

x_1	x_2	x_3	x_4	Amplitude
1	0	0	0	-8
1	0	0	1	-7
1	0	1	0	-6
1	0	1	1	-5
1	1	0	0	-4
1	1	0	1	-3
1	1	1	0	-2
1	1	1	1	-1
0	0	0	0	1
0	0	0	1	2
0	0	1	0	3
0	0	1	1	4
0	1	0	0	5
0	1	0	1	6
0	1	1	0	7
0	1	1	1	8

The recorded 4-bit data distribution is shown in Fig. 2 where we separate the empirical distribution due to coded transmitted “one” bits and “zero” bits. There was a total of 1,752,001 samples used to obtain these distributions. This corresponds to signal-to-noise ratios from 3 to 5 dB. We did not obtain separate distributions for more limited ranges of signal-to-noise ratios.

In reducing the 4-bit data to 3-bit data the two experiments performed the following reductions:

Hardware decoder:

recorded 4-bit data				decoder input data		
x_1	x_2	x_3	x_4	\Rightarrow	x_1	x_3 x_4

Software decoder:

recorded 4-bit data				decoder input data		
if	$x_1 = x_2$		\Rightarrow	x_1	x_3	x_4
if	$x_1 = 0, x_2 = 1$		\Rightarrow	0	1	1
if	$x_1 = 1, x_2 = 0$		\Rightarrow	1	0	0

Here the decoder input data correspond to 8 uniformly spaced amplitudes as follows:

Decoder input bits			Amplitude
1	0	0	-4
1	0	1	-3
1	1	0	-2
1	1	1	-1
0	0	0	1
0	0	1	2
0	1	0	3
0	1	1	4

Note that the only difference in these two coded data reductions appears when $x_1 \neq x_2$. The fraction of data samples where this occurred was measured and found to be

$$Pr\{x_1 \neq x_2\} = 0.30678 \quad (1)$$

When $x_1 \neq x_2$, the hardware decoder reduction “folds over” the 4-bit data by converting large amplitudes of the 4-bit data to 3-bit data amplitudes as follows:

4-Bit Data Amplitude		3-Bit Data Amplitude
-8	\Rightarrow	-4
-7	\Rightarrow	-3
-6	\Rightarrow	-2
-5	\Rightarrow	-1
5	\Rightarrow	1
6	\Rightarrow	2
7	\Rightarrow	3
8	\Rightarrow	4

The resulting 3-bit hardware decoder input amplitude distributions are shown in Fig. 3. The software decoder reduction essentially truncates the large 4-bit data amplitudes so that amplitudes -8, -7, -6, and -5 of the 4-bit data are converted to amplitude -4 of the 3-bit data. Similarly, amplitudes 8, 7, 6, and 5 are converted to amplitude 4. These 3-bit software decoder input amplitude distributions are shown in Fig. 4.

III. Analysis

For any binary input channel which has output denoted y we can define probability distributions $P_+(y)$ and $P_-(y)$ corresponding to a "zero" channel input bit (+) and a "one" channel input bit (-) respectively. For any such channel and the $\nu = 7$, $r = 1/2$ convolutional code, the decoded bit error probability P_b is bounded by (Ref. 2)

$$P_b \leq \left. \frac{\partial T(D, I)}{\partial I} \right|_{I=1, D=D_0} \\ = 36 D_0^{10} + 211 D_0^{12} + 1404 D_0^{14} + 11633 D_0^{16} \\ + \text{higher powers of } D_0 \quad (2)$$

where

$$D_0 = \sum_y \sqrt{P_+(y)P_-(y)} \quad (3)$$

First without any quantization, when "zero" is sent over the white Gaussian noise channel the matched filter output is assumed to be a Gaussian random variable with mean $A = \sqrt{E_s} = \sqrt{E_b/2}$ and variance $\sigma^2 = N_0/2$. When "one" is sent the mean is $-A$. For this unquantized case we have

$$P_+(y) = \frac{1}{\sqrt{\pi N_0}} e^{-\frac{(y-A)^2}{N_0}}; \quad \text{all } y$$

$$P_-(y) = \frac{1}{\sqrt{\pi N_0}} e^{-\frac{(y+A)^2}{N_0}}; \quad \text{all } y \quad (4)$$

and

$$D_0 = e^{-\frac{A^2}{N_0}} \\ = e^{-\frac{E_b}{2N_0}}. \quad (5)$$

For the 4-bit data distribution shown in Fig. 2 we obtained directly

$$D_0 = \sum_y \sqrt{P_+(y)P_-(y)} \\ = 0.245176 \quad (6)$$

Using the bound² in (2) we have for this 4-bit data

$$P_b \leq 4.42 \times 10^{-5}. \quad (7)$$

Similarly, for the 3-bit data of the hardware decoder experiment given in Fig. 3, we have directly

$$D_0^* = 0.332164 \quad (8)$$

and the bound from (2)

$$P_b^* \leq 1.50 \times 10^{-3}. \quad (9)$$

Finally, with the software decoder experiment of Fig. 4,

$$D_0^{**} = 0.250502 \quad (10)$$

and

$$P_b^{**} \leq 5.61 \times 10^{-5}. \quad (11)$$

The bound given by (2) is known to be reasonably tight, and so we show the bounds in Fig. 1 as P_b , P_b^* , and P_b^{**} . Because of uncertainty as to the signal-to-noise ratio, E_b/N_0 , corresponding to the total distributions of Figs. 2, 3, and 4, we used the midvalue of 4 dB for the signal-to-noise ratio. It is clear that the large difference between the hardware decoder experiment and the software decoder experiment can be explained by the foldover effects observed in reducing the original 6-bit data to 3-bit data in the hardware decoder experiment. Finally, note that P_b^* appears to be slightly smaller than expected. This can be accounted for by the fact that higher power terms in

²We ignore terms of powers of D_0 greater than 17.

the bound in (2) contribute non-negligible amounts to P_b^* for $D_0^* = 0.332164$. We took only the terms to the 16th power of D_0^* in the above bound. Ignoring these higher order terms for $D_0 = 0.245176$ and $D_0 = 0.250502$ makes little difference in the bound on P_b and P_b^{**} , respectively.

IV. Discussion

The MVM73 experiment included both X-band and S-band data. The S-band data also consisted of convolutionally coded bits represented by the same periodic sequence 1 1 1 0 1 0. For S-band, the signal-to-noise ratio was too low to show any significant coding gain. In reducing the original 6-bit data to 3-bit decoder input data, we find that for the S-band data

$$Pr\{x_1 \neq x_2\} = 0.13566 \quad (12)$$

This means that for S-band there was much less “fold-over” observed in the hardware Viterbi decoder input data. Indeed, if we were to observe some sections of the MVM73 S-band data, we could easily conclude that

dropping x_2 , as was done in the hardware decoder data, would make little difference in decoder performance.

Finally, we note that in reducing 6-bit data of the form $x_1, x_2, x_3, x_4, x_5, x_6$ to 3 bits, an obvious suggestion is to take the 3 most significant bits x_1, x_2, x_3 . If this is done for the X-band data, we get the data distributions shown in Fig. 5 and the resulting parameter

$$D_0^{***} = 0.2644 \quad (13)$$

with bit error bound from (2) of

$$P_b^{***} \leq 1.03 \times 10^{-4}. \quad (14)$$

While this data reduction yields better decoder performance than the hardware data reduction which caused foldover effects, it is not as good as the software decoder data reduction. The Viterbi decoding performance seems to be sensitive to the probability distribution values for small amplitudes, and taking the 3 most significant bits for the decoder input data yields too coarse a division of the input amplitudes.

Acknowledgment

The authors thank J. C. Springett, W. J. Weber, and J. W. Layland of the Jet Propulsion Laboratory for their contributions to this work.

References

1. Heller, J. A., and Jacobs, I. M., “Viterbi Decoding for Satellite and Space Communication,” *IEEE Trans. Commun. Technol.*, Vol. COM-19, pp. 835–848, Oct. 1971.
2. Viterbi, A. J., “Convolutional Codes and Their Performance to Communication Systems,” *IEEE Trans. Commun. Technol.*, Vol. COM-19, pp. 751–772, Oct. 1971.

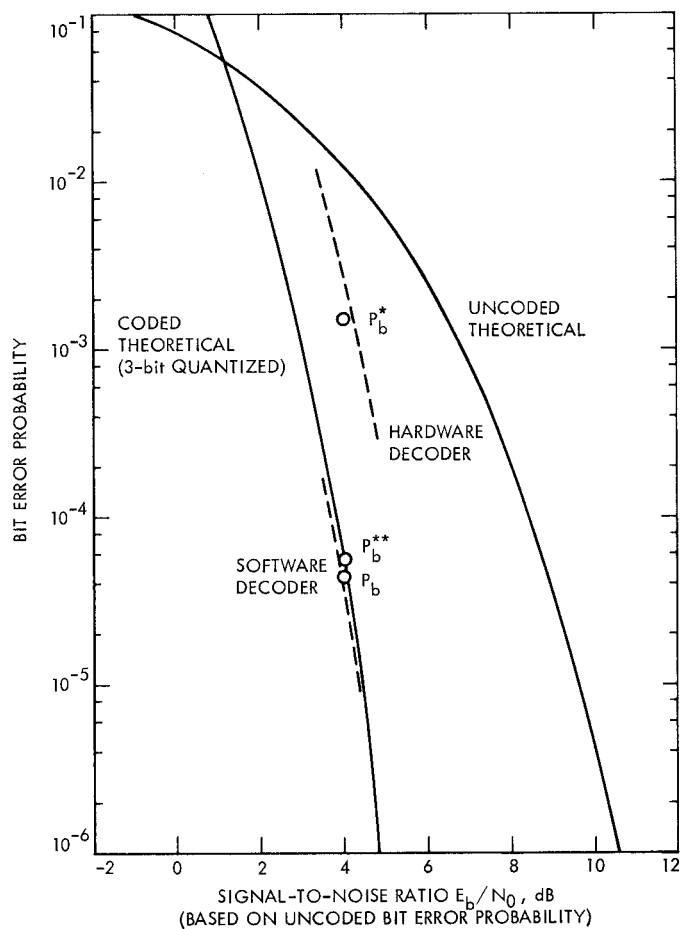


Fig. 1. X-band decoded bit error rates

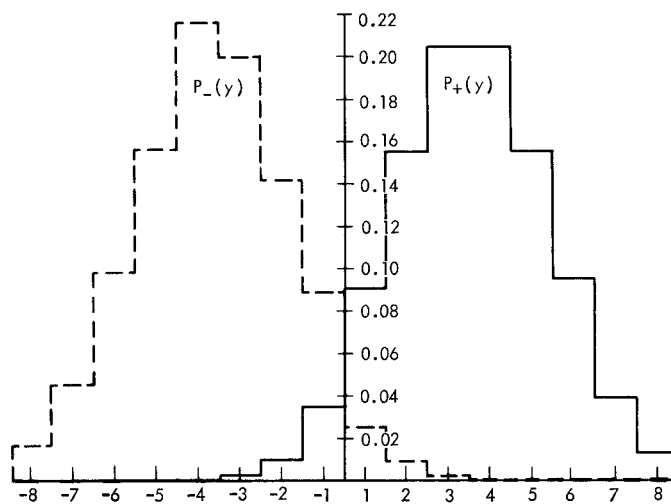


Fig. 2. 4-bit data distribution

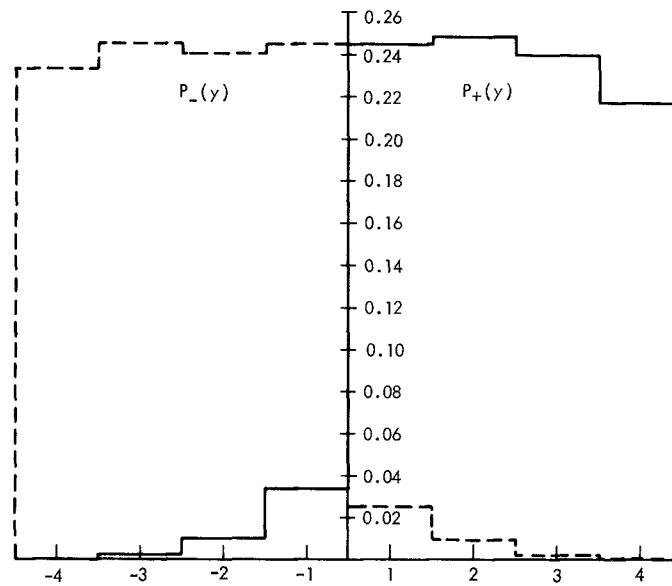


Fig. 3. 3-bit hardware decoder data

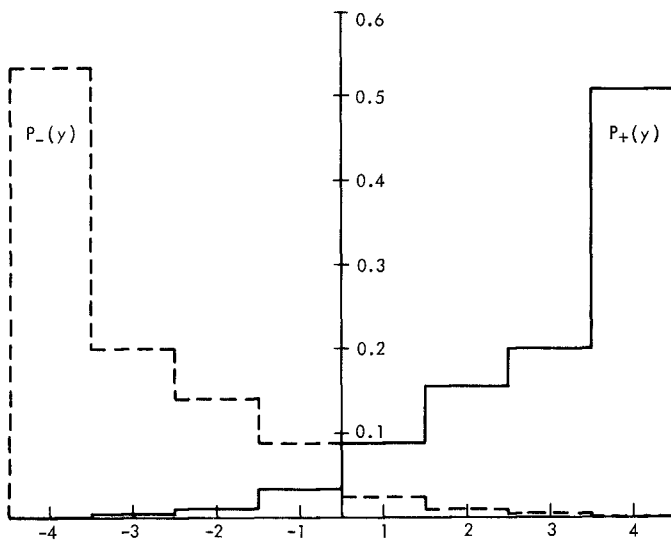


Fig. 4. 3-bit software decoder data

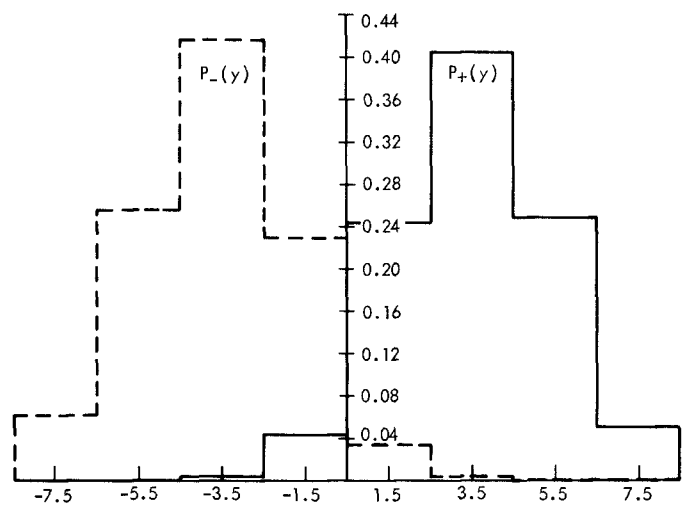


Fig. 5. 3 most significant bits